Machine generated alternative text:
Source System
Kafka Cluster
+ topics
+ partitions :
+ replications ;
+ partition leader & in-sync-replicas (ISR)
+ offsets topic
f I
+ round robin
+ key based ordering
+ acks strategy
+ consumer groups !
+ at least once !
+ at most once !
Broker 101
---,
Producers
---.
Broker 102
I..
---.
Consumers
Broker 109
---.
Target Systems

* 1. Topics
  2. Partitions
  3. Offsets
  4. Producers
  5. Consumers
  6. Consumer Groups
  7. Message Keys
  8. Topic Replication
  9. Producer acknowledgements
  10. Topic Durability
  11. Zookeeper
  12. Kafka Kraft

DEV control center : <URL>

It's a Magic Table => Without any constraints

* You can Write data to Topic
* You can read data from Topic
* You can't write queries to read data

* **Topics: a particular stream of data**
* Like a table in a database (without all the constraints)
* You can have as many topics as you want
* A topic is identified by its name
* Any kind of message format
  + JSON, avro, binary, ProtoBuf
  + Use **Avro** if you need schema evolution and strong data governance.
  + Use **Protobuf** for compact and efficient messaging with strong typing.
  + Use **JSON** for flexibility and ease of use.
  + Use **MessagePack** for optimized performance.
  + Use **Raw Byte Arrays** for maximum control over serialization

* The sequence of messages is called a data stream
* You cannot query topics, instead, use Kafka Producers to send data and Kafka Consumers to read the data